

CI: Push Linux Build to Launchpad

Initiate the push with 'git push origin [tag]'

In .travis it is possible to initiate the push of a Linux build to your Launchpad repository. These instructions assume that you have a travis account and know your way around the services and basic process. Your travis account is used to confirm 'successful' builds of linux and osx, and to create Linux packages to deploy to your Launchpad.

How to push a .Travis 'built' file to your Launchpad

For travis refer to [Petri's post #637](#)

A. Configure the Travis file for security and permissions

1. You can use this OCPN_Draw_pi [.travis](#) file as an example. The important part is the “- provider: launchpad” section.

B. Launchpad provider at bottom of travis.yml file

Example Bottom section of the travis.yml file. Note: indentation is important in yml files, child parts MUST be indented below the owner. This is shown below by “oauth_token:” being the owner and “secure:” being the child, notice that “secure:” is indented.

```
deploy:
  - provider: launchpad
    slug: "Your repository slug here"
    oauth_token:
      secure: "Your encrypted oauth_token here"
    oauth_token_secret:
      secure: "Your encrypted oauth_token_secret here"
    skip_cleanup: true
    on: # Set Deploy conditions
      # Deploy only when tag is not specified
      # tags: false
      # or this branch
      #branches:
      #  only:
      #    - master
      tags: true
      all_branches: true
```

```
condition: "$TRAVIS_OS_NAME = linux"
```

The "slug:" and "secure:" data items are in double quotes ("").

If you want another deploy section, like git, you will need to create a new "- provider:" with the text "releases" following it. Both "- provider:" labels need to have the same indentation.

C. Install Travis CLI to encrypt a Launchpad secure public key

1. You need a travis CLI (command line interface) to be able to encrypt the Launchpad tokens. The encrypted strings are put in your personal .travis.yml file.
2. Travis CLI installation instructions for Windows. When you are sure that you have travis CLI up and working, you can start to create the encrypted key. Without it, deployment to GitHub does not work.
3. Travis CLI installation instructions for Linux. Windows10 users can create a [WSL-Windows Subsystem for Linux \(Ubuntu 18.04LTS\) on Win10 Installation](#), and then from the linux command prompt execute:

```
sudo apt-get update --fix-missing
sudo apt-get install ruby
sudo apt-get install gem
ruby -v
gem install travis -v 1.8.10 --no-rdoc --no-ri
travis version
response: 1.8.10.

225mb disk space used.
```

D. Install "curl" to allow posting information to Launchpad

Part of the process requires the http "Post" process to acquire security tokens. The easy way to do this from a command prompt using the "curl" program. * For Linux: Your package manager should be able to install "curl" as it is a standard internet tool. * For Windows: goto this website <http://techieroop.com/how-to-install-curl-on-windows-10/> and select the correct version for your setup.

E. Create a Launchpad "request-token"

Issue the following command from a command prompt.

```
curl --dump-header - -H "Content-Type: application/x-www-form-urlencoded" -X
POST --data
'oauth_consumer_key=Travis+Deploy&oauth_signature_method=PLAINTEXT&oauth_signa
```

```
ture=%26' https://launchpad.net/+request-token
```

You should receive a reply like:

```
HTTP/1.1 200 Ok
Date: Sat, 06 Jul 2019 01:57:06 GMT
Server: zope.server.http (HTTP)
X-Powered-By: Zope (www.zope.org), Python (www.python.org)
Content-Length: 132
X-Xss-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=15552000
Vary: Cookie,Authorization,Accept-Encoding
X-Content-Type-Warning: guessed from content
X-Launchpad-Revision: 19007
X-Frame-Options: SAMEORIGIN
Content-Type: text/plain; charset=utf-8

oauth_token=XJwTXv9f8RRRT6TwCZ4x&oauth_token_secret=lbrXHsQ0jVlJlqLpNX4mGwtNjc
Rp2rWzqhQx53KDhTM87m7f7MMGTNw7DsG3CcbhSbn3jwTCmMkFbFCG
```

Save this information as you will need it in following steps.

F. Authenticate the User

You will now need to logon to your Launchpad account. You need to have been authenticated before you can issue the next command from your browser. Open a new browser tab and enter the following in the “Address line”:

```
https://launchpad.net/+authorize-token?oauth_token={your oauth_token from step D:}
```

Your version could look like:

```
https://launchpad.net/+authorize-token?oauth_token=XJwTXv9f8RRRT6TwCZ4x
```

You will now be asked for the level of access to provide this token. The easiest is to select “All”, the last button on the page.

G. Get new secure Launchpad token

In your command prompt window issue the following command:

```
curl --dump-header - -H "Content-Type: application/x-www-form-urlencoded" -X
```

```
POST --data 'oauth_signature=%26{your
oauth_token_secret}&oauth_consumer_key=Travis+Deploy&oauth_token={your
oauth_token}&oauth_signature_method=PLAINTEXT'
https://launchpad.net/+access-token
```

Your version would look like“

```
curl --dump-header - -H "Content-Type: application/x-www-form-urlencoded" -X
POST --data
'oauth_signature=%26lbrXHsQ0jVlJlqLpNX4mGwtNjcRp2rWzqhQx53KDhTM87m7f7MMGTNw7Ds
G3CcbhSbn3jwTCmMkFbFCG&oauth_consumer_key=Travis+Deploy&oauth_token=XJwTXv9f8R
RRT6TwCZ4x&oauth_signature_method=PLAINTEXT'
https://launchpad.net/+access-token
```

From this you would get a response like:

```
oauth_token=DJZmQkJ65F3M0HD4Lws5&oauth_token_secret=WbwMzlDhW3HpkHxLh7wrtkXqPZ
kFS8n6HHXrZXLj4WK0vKljJqHtdx9hWmCLXJLZ0CQSc5kD8jWnsdn&lp.context=None
```

G. Encrypt your tokens before use

You should now encrypt your tokens before they are used. This makes the interaction more secure.

- Change directory to your local code base for git.
- Issue command:

```
travis encrypt {your oauth_token from step F.> --org
```

- This may look like:

```
travis encrypt DJZmQkJ65F3M0HD4Lws5 --org
```

- You will receive a response like:

Please add the following to your .travis.yml file:

```
secure:
"0+F3qadTm1EojsoGlopwLPirEJ8TkshlyZG0Ud0a62YJsI7SrE3ugryeU2wuYHRBxE0dhXDCxHkUf
+E45yq4PL8IIHx29eoygsZeVwHCgyT2nN/Pk/0/E0JXeCRo5WgrliyDnHCL4Dmz7vyRiP6svWuzz1x
M6ATpKXSULb02tdzcoydD0bzeUYwUnK2VRc7yv4Afxgr0yXx+7fdscV+ALCZ2J0vroxXWGRJawo96K
WEE00n72/wiYZXeMuC7QtxFBQopSRNrmzGMTHv9F8c6cGT/Utcpr86xw42J00px0XCdd5i+60hzlA6
QWswrNIuNBVnYwydGfJoX56bcalRZ1MEE+NyU/zHYBi8uehh8NIfUAYxdY7w7VW6FtyEdayNGWBm1F
RGyy0quYfFQMGMHCBBrnt30IKk2glSFWLxUBUTec4c3GG0X08W0Zq/IhGE2ej77AVzCXGb1z0iruUSJ
dpkpQH73xC174SQBmiW54k76V4oRy2YyIXeibC04zKVhSpfJ00R0bvbZbz4LAgx0ML4emxFUFsIkzC
NK872rEPDiUJjcl6So3bNKTrV0RC44uXN+JzFXrAeHwCZVATXANg4Ah4N94YX4W5sk0iNArhN6/p50
kZbffxvZnYlPIh728BvtCFkGi5IELEeFt0Q/wE1BUr6k7ozxhGJtSTohaDb0="
```

Pro Tip: You can add it automatically by running with `--add`.

- PS. Don't do as suggested with the `-add` as it may destroy your indentation in your `.travis.yml` file. It is far better to select the line starting with `"secure:"` and copy paste it into the requisite position in the `.travis.yml` file, in this case under `"oauth_token"`
- Issue command:

```
travis encrypt {your oauth_security_token from step F.> --org
```

- This may look like:

```
travis encrypt
WbwMzldHw3HpkHxLh7wrkXqPZkFS8n6HHXrZXLj4WK0vKljJqHtdx9hWmCLXJLZ0CQSc5kD8jWns
dn --org
```

- You will receive a response like:

Please add the following to your `.travis.yml` file:

```
secure:
"zfcZagTCDKN2PrwAW4jHDMFJ40EhDwscLYguiv0j+MlrI9A/Cs1v6IDjHa3Rm/v3elZuZMkZ10rZT
8LUIzKaJS8MsMMU1jFnscAsqW8422S2chJLVN908T/rd0QinHotJdk1E7NbSLgHQF5J0eQ31I50kq4
xWNhZ0Qe9G9FVEKi+f8HG9M3idJn00iTJeaMqlgi6hj6X6ajgv9kaKe5qDGAwn8hB7nhav8epNCgw6
zqgygKg3cTm6WF/KMLqYc6BrHruU+e3y0An9fBMD8TI0mRgHWexwkvB0kMjBZAb06n3hJL+cXEiUI
7d+jdY9HirwX40jWTBXKE3jvF3SaWkqZGLj8/rRWq1TPkC8GwnDBLfiukmW2x+a982L0pxzDN5vs1X
lzGMys5GzIc0TGaHsKd/BAMRXAQavcmTp9Zm51XsXismdD+CHCNm7BV7f8RaT0gWRw0+JLgT1wCsRI
WoYsIVO67K2NYxgh+K3K36/8n/RjZ0y0id/+zsPZh0RKsmRThFqx09kT00Zyi0yn6aqGks2x0GAq98
BYdKiRxRlQquKk+daXhvsDCGvLCnC+8ogjvVDXSkt9bXNDW6qEcpdp4Y7v2in+K0GR1DBGXx75rB13
h39ZLaeYKmnBXMncsRVYHk3MkF8mYSnu2DX/t50um/TFYwLr0b485I5tKcf4="
```

Pro Tip: You can add it automatically by running with `--add`.

- PS. Don't do as suggested with the `-add` as it may destroy your indentation in your `.travis.yml` file. It is far better to select the line starting with `"secure:"` and copy paste it into the requisite position in the `.travis.yml` file, in this case `"oauth_token_secret"`

H. Start building the `.travis` project

1. You will need to turn the project "on" after signing into your travis account. Click on the icon in the top right with the down chevron and select 'Settings', the page will display your repositories under the 'Repositories' tab. Make sure the ones you want to use have the slider button to the right with a tick in the button. If it is to the left and shows a cross it is not active.
2. Then when you commit `.travis.yml` file, the travis build will start. It is good to check that both of its build processes do pass faultlessly before the next step.
3. It takes awhile for each commit (Linux and Mac) to build correctly, be patient when it is running.
4. Make sure it built successfully. The project will not be saved because travis does not provide that type of storage space.

I. Troubleshooting

When the Travis build does not work the error messages can be obscure and not really help too much. A typical message may be:

Skipping a deployment with the provider because this branch is not permitted:

This is not very helpful as it just says something went wrong not what.

1. Check your whole .travis.yml file. Look at it carefully and compare it to the example given.
2. Make sure you have the security token enclosed in double quotes, i.e. "security/token"
3. Make sure the build process has actually worked for the file types you are trying to transfer to github
4. If you have extra values in your yml file make sure you really know what they do. If you have issues, take all the extraneous values out and see if it works then.
5. If necessary find a very small project and play with that as it will compile much quicker.
6. Make all changes in a branch of the main repository as you will have to do multiple commits to get it working. Once it is working you can make the changes to your main branch(es) and clean up the 'test' branch

:

From:
<https://opencpn.org/wiki/dokuwiki/> - **OpenCPN Manuals**

Permanent link:
https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:ci-push-linux-build-to-launchpad

Last update: **2020/03/09 00:15**

